

Systems [1.0]

1. [1.0 points] When defining a complex system using a systemic approach, one principle is that a system can be broken into smaller subsystems and these subsystems interact between each other using interfaces.

- a) Present two distinct types/classes of interfaces that sub-systems can implement, and that the programmer can use to link those sub-systems.
- b) For each type of interface, present the scope of usage (local/remote) and the major concerns that should be considered when implementing and using such interfaces.

Concurrency [2.0]

2. [1.0 points]

- a) Describe what is Concurrency.
- b) Describe how concurrency can be implemented in an Operating System with only one CPU

3. [1.0 points]

- a) Present and describe two types of parallelism.
- b) For each one, present one example of an implementation of such type.

Threads / Synchronization [3.0]

4. [1.0 point] Suppose that a program reads lines from a file and for each line creates a new thread immediately after reading it.

- a) Write the C/pseudo code that reads text lines from a previously opened file and creates a new thread for each line immediately after reading it. Each thread should receive as argument a different line from the file.
- b) Write the code inside the thread that prints the received line length.

Guarantee that the relevant arguments for `pthread_create` and the thread code are correct.

5. [1.0 point]

- Explain what happens to the program (and other threads) when a thread erroneously terminates before unlocking a lock (inside a critical region).

6. [1.0 point] Multiple threads can access global arrays without requiring synchronization if each thread accesses different array positions.

- a) Describe one problem that may happen when different threads access distinct positions on the same array.
- b) Explain in detail the type/pattern of accesses and the consequences to the program execution.

System calls / Processes [2.0]

7. [1.0 point] System calls are different from the regular C libraries function.

- a) Describe how a system call is executed from the moment it is called.
- b) Give one example of one system call and one C library function that have a similar objective. Explain how they are related and different.

8. [1.0 point] Processes were the first mechanism that implemented concurrency and parallelism in modern operating systems, but are still used today when implementing servers.

- Describe why most mail servers running on Linux create a new process after receiving the connection of a client, instead of creating a new thread.

Inter-process communication (IPC) [2.0]

9. [1.0 point] Shared memory is one of the types of IPC mechanisms, sockets (datagram and stream) are another.

- a) Present one other IPC mechanism that allows communication between processes in the same machine.
- b) Describe one difference between the presented IPC mechanism and shared memory.
- c) Describe one difference between the presented IPC mechanism and sockets.

10. [1.0 point] In the project, students changed datagram sockets into stream sockets when implementing the new-super-pong.

- a) Describe two advantages of using stream sockets over the use of datagram sockets.
- b) Present one major change in the organization of a program when changing the communication from data-gram sockets to stream sockets.